Scuola universitaria professionale della Svizzera italiana
Dipartimento tecnologie innovative

SUPSI

Viki: Smart Home Natural Language interface

Relatore

Nicola Rizzo

Studente/i

Luca Ambrosini

Correlatore

Alan Ferrari

Committente

-

Corso di laurea

-

Modulo

M00002 Progetto di diploma

Anno

2015/16

Data

11 luglio 2016

Capitolo 1

Introduzione

Disegnare una macchina in grado di comportarsi come un umano, in particolare di parlare e interpretare il linguaggio, è uno degli obbiettivi dell'ingegneria sin da metà del 20esimo secolo. Le interfacce in linguaggio naturale sono considerate come il punto di arrivo dell'interazione uomo macchina. Lo sviluppo in questo campo è stato molto intenso negli ultimi anni ciò ha permesso la realizzazione di agenti intelligenti, che simulino una conversazione con la persona e che riescano a compiere azioni più complesse di semplici comandi con frasi standardizzate.

1.1 La voce come mezzo di comunicazione

1.2 Cenni storici

Il primo esempio nella storia di dispositivo ad interazione vocale è da collocare nell'estate del 1952, presso i laboratori Bell. Quell'anno vennero eseguiti i primi test di "Audrei" (Automatic Digit Recognizer), un dispositivo in grado di comporre un numero di telefono dettato ad un microfono.

Nel 1962 IMB presentò "Shoebox", una macchina in grado di comprendere 16 diverse parole pronunciate in inglese. Questa macchina era destinata ad essere una calcolatrice vocale.

Lo sviluppo di sistemi in grado di comprendere il linguaggio naturale è poi proseguito nel tempo, passando dalla comprensione di pochi suoni alla comprensione continua del linguaggio naturale; le tecniche si sono evolute passando da metodi statistici fino ad approcci basati sul deep learning. Esso è una branchia del machine learning, che simula delle reti neurali multi strato che riescono ad apprendere funzioni complesse. [1]

Grossi miglioramenti in questo campo sono pervenuti nell'ultimo secolo, soprattuto grazie all'incremento delle capacità computazionali. Questo ha permesso la realizzazione di agenti intelligenti sempre più complessi.

1.3 Evoluzione degli agenti

I primi dispositivi ad interazione vocali sono gli "Interactive Voice Response", cioè gli agenti dei call center, che descrivono attraverso la voce i comandi e ricevono input attraverso i numeri digitati sul telefono. Il numero di input era quindi molto ridotto e la struttura della conversazione era fissa.
Successivamente i lettori automatici e i dispositivi ad interazione vocale sono stati integrati nei sistemi operativi. La loro funzione principale consisteva nell'aiutare le persone con delle disabilità. Era comunque necessario un microfono, quindi una prossimità al computer. Inoltre la voce aveva una funzione di sostituzione delle capacità visive o motorie non erano previste funzionalità dedicate che permettessero una maggior produttività.
Con l'ultima generazione di smartphone, che sono dotati di un microfono e che dispongono di una connessione a Internet, gli agenti intelligenti sono diventati parte della nostra vita quotidiana. Vista la limitata capacità di calcolo degli smartphone tutto il processamento dell'informazione viene eseguito attraverso cloud computing, che utilizza tecniche di deep learning.
L'ultima generazione di dispositivi ad interazione vocale è costituita da "Amazon Echo" e "Microsoft Kinect", essi sono in grado di ricevere input vocali in modo continuo, senza che l'utente debba avere un microfono addosso e senza che venga azionato un dispositivo. Questo ha portato l'interazione vocale a un nuovo livello di usabilità e ha aperto nuove possibilità di utilizzo di questa tecnologia nell'ambito delle smart home.

1.4 Il momento giusto

Storicamente lo scetticismo a proposito delle interfacce in linguaggio naturale è sempre stato molto elevato: soprattuto per la loro scarsa produttività sono sempre state considerate un accessorio e non una tecnologia che potesse essere sfruttata. Ora però tutte le tecnologie necessarie alla realizzazione di un agente intelligente che ci possa aiutare nella vita quotidiana sono pronte:

- **Speech-To-Text**: Negli ultimi anni, soprattuto grazie alle tecniche di machine learning, questa tecnologia è arrivata ad alti livelli di accuratezza, superando in alcuni casi perfino le capacità di percezione dell'uomo. So-

no ormai disponibili componenti che eseguono speech-to-text in tutte le lingue del mondo.[2]

- **Comprensione del testo**: L'analisi semantica, la vettorizzazione di parole e frasi, permettono una sempre maggior strutturazione del contenuto del testo, la quale consente una migliore comprensione da parte delle macchine.[3]
- **Connessione**: La capacità di calcolo richiesta per effettuare STT e comprendere un testo è molto elevata, per questo in genere si ricorre a un server remoto; l'incremento della larghezza di banda e la diminuzione dei tempi di latenza hanno reso possibile delle risposte in tempi adeguati.
- **Audio always on**: La tecnologia ha permesso la creazione di dispositivi che ascoltano in modo continuo e sono in grado di riconoscere delle keyword per la loro attivazione ("Ehi Siri"), le persone inoltre si sono abituate e hanno imparato ad accettare questa profonda invasione della privacy.
- **IOT**: Si stima che il mercato dell'IOT raggiunga una cifra d'affari di 1200 Miliardi di $ entro il 2020 e l'home automation è uno dei settori nei quali un agente può raggiungere la sua massima utilità.

Capitolo 2

Caso d'uso

L'utilità degli Agenti Intelligenti ad interazione vocale è spesso messa in dubbio, ma ci sono alcune occasioni nelle quali le loro capacità brillano, poiché forniscono un'esperienza d'uso diversa dalle interfacce basate su schermi o touch.

- **Accessibilità**: Consentono un'esperienza d'uso soddisfacente a persone con disabilità motorie o visive, in quanto la procedura di descrizione delle operazione possibili e la successiva richiesta di un input non è più necessaria, gli agenti possono infatti eseguire comandi in risposta a frasi come "Manda un messaggio a Mario dicendo che arriverò tardi"
- **Eye-busy o Hand-busy**: In scenari quali la guida o attività svolte in cucina, in cui si hanno le mani impegnate e non si ha la possibilità di concentrare la propria attenzione su uno schermo, gli Agenti Intelligenti diventano particolarmente utili.
- **Automazione casalinga**: Supportare la creazione di comandi complessi a discrezione dell'utente, che permettano di compiere azioni in modo semplificato. Ad esempio "Buonanotte" potrebbe automaticamente abbassare tutte le tapparelle e spegnere tutte le luci.

In queste situazioni sarebbe quindi ideale avere un agente in grado di svolgere per noi la maggior parte delle operazioni che gli vengono indicate attraverso la voce, come se stessimo conversando con una persona alla quale chiediamo di svolgere il compito.

Capitolo 3

Obbiettivo

Il progetto è basato su "Viki", un Agente Intelligente[4], capace di controllare molti degli apparecchi presenti in un abitazione e di fornire informazioni, ad esempio riguardanti il meteo. Esso è stato sviluppato presso l'Istituto Sistemi Informativi e Networking. Il primo obbiettivo del progetto di bachelor consiste nella comprensione dell'infrastruttura del sistema attuale; successivamente si vuole migliorare l'interazione vocale con il sistema, cercando di renderla il meno rigida possibile. Inoltre si provvederà all'estensione delle API disponibili e si implementeranno strutture che miglioreranno l'intelligenza dell'agente attuale.

3.1 Interfaccia in linguaggio naturale

3.1.1 Grammatiche fisse

Il sistema attuale prevede l'interazione vocale, ma utilizza un sistema basato su delle grammatiche fisse. Questo implica quindi una struttura della frase definita a priori dal programmatore, che nel caso non sia rispettata, impedisce la comprensione del comando da parte dell'agente.

3.1.2 Rimozione dei vincoli

Il progetto aspira a creare un interfaccia libera da questi vincoli, che provi a comprendere il senso della frase in modo indipendente dai singoli vocaboli e dalla struttura utilizzata. Grazie all'interfaccia libera l'utilizzatore può concentrarsi sull'azione da eseguire e meno su come esprimerla per far si che l'agente sia in grado di comprenderla. Una delle critiche che viene più spesso mossa alle interfacce in linguaggio naturale è la necessità dell'utilizzatore di compiere uno sforzo mentale per pensare come la macchina. Grazie alla rimozione di questi vincoli l'utilizzatore dovrebbe trovare l'interazione con l'agente più simile

a una conversazione tra persone, garantendo quindi una maggior soddisfazione. Un'interfaccia di questo livello semplificherebbe l'utilizzo di una smart home al punto di renderla fruibile anche a persone che non si trovano normalmente a loro agio con la tecnologia.

3.2 Incremento delle API

3.2.1 API attuali

Le capacità del sistema sono strettamente collegate alla mole di informazioni alle quali esso ha acceso e ai dispositivi che è in grado di controllare. Al momento Viki può controllare :

- Lampadine philips HUE (accensione, colorazione, intensità)
- Prese di corrente z-wave (accensione, lettura potenza istantanea)

e ha accesso alle seguenti informazioni:

- Sensori di movimento, luminosità, umidità, temperatura
- Previsioni meteo (yahoo)

3.2.2 API future

Durante lo sviluppo del progetto di bachelor si vogliono incrementare le capacità del sistema, in particolare Viki dovrà essere in grado di controllare:

- Tapparelle motorizzate
- Mediacenter
- Impostazione di sveglie
- Impostazione di promemoria
- Aggiunta eventi calendario
- Impostazioni timer

e avrà accesso a informazioni aggiuntive quali :

- Palinsesto televisivo (RSI, Mediaset, Rai)

3.3 If then else

Sviluppo futuro

Capitolo 4

Comunicazione engine - voice

Il modulo di interazione vocale è realizzato come un componente esterno dal sistema di gestione dell'abitazione, cioè quello che si occupa di accedere alle informazioni e di azionare gli attuatori. E' stato quindi necessario definire un protocollo che informasse il sistema di controllo vocale di quali operazioni possono essere compiute e quali tipologie di informazioni sono disponibili.

4.1 Struttura dell'informazione

Per definizione l'insieme delle operazioni che il sistema di gestione è in grado di compiere abbiamo definito la seguente struttura:

- **Universe**: l'insieme di tutti i domini.
- **Domain**: un oggetto o un dominio di informazione che il sistema rende disponibile, per essere azionata o interrogata (es. Lampada, Tapparella, Meteo, Palinsesto)
- **Operation**: sono definite nell'ambito di un dominio e rappresentano le operazioni che possono essere richieste (es accensione di una luce, richiesta delle previsioni metereologiche)
- **Parameters**: sono definiti nell'ambito di un operazioni e rappresentano i parametri che possono essere associati a un operazione (es. colore da impostare per la lampada, luogo per le previsioni metereologiche)
- **ParameterType**: i parametri precedentemente definiti devono essere di una tipologia specifica(es. Data, Luogo)

4.1.1 Tipologie di parametri

Il sistema supporta parametri tipizzati, che possono appartenere alle seguenti categorie:

- LOCATION
- DATETIME
- NUMBER
- COLOR
- FREE_TEXT

4.2 Formalismo

Per la comunicazione della struttura precedentemente definita tra l'agente intelligente e l'interfaccia vocale si è scelto di utilizzare il formato JSON

4.2.1 Universe

Nome	Descrizione	Tipo
id	Identificativo univoco	String
domains	Lista dei domini che compongono l'universo	JSONArray di Domain

Tabella 4.1: Struttura JSON Universe

4.2.2 Domain

Nome	Descrizione	Tipo
id	Identificativo univoco	String
words	Parole associate al dominio (es. light,lamp)	JSONArray di String
friendlyNames	Nomi associate al dominio (es. "palla" -> lampada)	JSONArray di String
operations	Operazioni che possono essere eseguite nel dominio	JSONArray di Operation

Tabella 4.2: Struttura JSON Domain

4.2.3 Operation

Nome	Descrizione	Tipo
id	Identificativo univoco	String
words	Parole associate al dominio (es. light,lamp)	JSONArray di String
textInvocation	Frasi per invocare l'operazione	JSONArray di String
mandatoryParameters	Parametri obbligatori per l'operazione	JSONArray di Parameter
optionalParameters	Parametri opzionali, non necessari	JSONArray di Parameter

Tabella 4.3: Struttura JSON Operation

4.2.4 Parameter

Nome	Descrizione	Tipo
id	Identificativo univoco	String
type	Tipo del parametro	ParameterType

Tabella 4.4: Struttura JSON Parameter

4.3 Modalità di comunicazione

Per trasmettere l'informazione precedentemente descritta abbiamo scelto di utilizzare un canale di comunicazione WEB, seguendo l'architettura REST.[5]
Il sistema permette quindi di reperire l'informazione attraverso una chiamata GET all'indirizzo *"/cose"*; la risposta consiste nel JSON precedentemente descritto, un esempio è presente in appendice a questa documentazione.

Capitolo 5

Comunicazione voice - engine

Il software di gestione vocale si occupa di estrarre i comandi che l'utente ha richiesto al sistema. Dopo aver completato il processamento dell'informazione restituisce la serializzazione in formato JSON di un oggetto di tipo Command.

5.1 Command

L'oggetto restituito rappresenta il comando che deve essere eseguito dal sistema, include inoltre la frase che l'utente ha pronunciato e la frase che nel sistema è associata al comando riconosciuto.

5.1.1 Struttura JSON Command

Nome	Descrizione	Tipo
domain	Id del dominio	String
operation	Id dell'operazione	String
said	Frase ascoltata	String
understood	Frase associata al comando nel sistema	String
paramValuePairs	Lista di parametri e relativi valori	JSONArray di ParamValuePair

Tabella 5.1: Struttura JSON Command

5.1.2 Struttura JSON ParamValuePair

Nome	Descrizione	Tipo
id	Id del parametro	String
type	Tipologia del parametro	ParamType
value	Valore assunto dal parametro	String

Tabella 5.2: Struttura JSON ParamValuePair

5.2 Modalità di comunicazione

Per la trasmissione dei comandi abbiamo scelto di utilizzare l'architettura REST. Il sistema è predisposto per l'esecuzione di comandi provenienti dall'esterno, offre un interfaccia all'indirizzo *"/sendCommand"*, attraverso il metodo POST. Al seguente indirizzo è possibile inviare comandi nel formato precedentemente descritto.

5.2.1 Risposta-feedback?

Reminder

Capitolo 6

Speech to text

Capitolo 7

Scelta della tecnologia

7.1 Linguaggio

python (spicy.io) vs Java (deeplearning4j + stanford nlp)

Capitolo 8

Ricerca dell'operazione

Dopo aver trasformato quanto detto dall'utente in testo, il sistema prova a mappare la frase su una delle azioni che che il sistema è in grado di eseguire. In particolare viene cercato un dominio di esecuzione dell'operazione (es. Meteo, Luci) e l'operazione che si vuole compiere in questo dominio (es. Ricerca condizioni metereologiche, accensione della luce). L'obbiettivo di questo sistema è di rendere generica l'interazione vocale, si è quindi dovuta trovare una metodologia che identificasse dominio e operazione indipendentemente dalle singole parole utilizzate, cercando quindi di astrarre il significato dai vocaboli. Il sistema per poter determinare l'operazione dispone di una lista di parole associate all'operazione, una lista di parole associate al dominio e opzionalmente una lista di frasi per l'invocazione dell'operazione in un dominio specifico.

8.1 Database lessicale : wordNet

Il primo approccio si è basato sull'utilizzo di wordNet, un database lessicale della lingua inglese. In wordNet ad ogni lemma è associata una definizione, come in un normale dizionario, ma i lemmi sono anche collegati da una serie di relazioni, formando un grafo. In particolare le relazioni utilizzate nel progetto sono antonimia, iperonimia, sinonimia, metonimia.[6]

Utilizzato questo strumento è possibile reperire i sinonimi di una parola, dividendoli per categorie grammaticali (es. sinonimi di light come aggettivo o come nome). Attraverso il confronto delle definizioni e delle relazioni è poi possibile determinare la similarità tra due vocaboli. Per definire tale metrica è possibile utilizzare molti diverse metodologie, tra i quali si è scelto di utilizzare quello definito come *path*. [7]

Questo approccio conta il numero di nodi che compongono il percorso più breve tra i due vocaboli, restituisce poi un coefficiente di similarità che corrisponde

all'inverso della distanza precedentemente calcolata. Due vocaboli il cui significato è molto simile avranno delle strette relazioni con parole simili, quindi un alto coefficiente di similarità.[8]

8.1.1 Calcolo della similarità

La similarità di un dominio con una frase viene calcolata come il massimo della similarità tra tutte le parole associate al dominio con tutte le parole presenti nella frase.
Lo stesso procedimento viene applicato alle operazioni e la similarità della coppia dominio/operazione è costituita dalla media aritmetica delle due similarità.

8.1.2 Vantaggi

Vantaggi:

- **Richiede poche risorse**: il database pesa meno di 100Mb.
- **Supporto ai phrasal verbs**: sono già compresi nel database come un singolo verbo (es. turn_off).

8.1.3 Svantaggi

- **Dizionario statico**: è stato creato manualmente e non viene aggiornato da anni.
- **Indipendente dalla frase**: il risultato dell'analisi di similarità è indipendente dall'ordine delle parole.
- **Dipendente dal pos**: per cercare i sinonimi solo nella corretta categoria grammaticale è necessario affidarsi al Part Of Speech tagger, la cui affidabilità non è totale.

8.2 Part-Of-Speech tagging

L'approccio precedentemente descritto presenta lo svantaggio di non tenere conto dell'ordine delle parole nella frase; si è quindi pensato di aggiungere un componente che tenga in considerazione questa informazione. Il Part Of Speech tagger è un componente che si occupa di assegnare a ogni parola un tag, si occupa inoltre di definire le relazioni che intercorrono tra questi tag.[9][10]

8.2.1 Calcolo della similarità

La prima fase consiste nella ricerca di un nome che corrisponde al dominio, da essa vengono poi seguite delle relazioni che legano il verbo a colui che compie l'azione o colui che la subisce. Nei verbi viene quindi cercata l'azione con l'approccio di path similarity attraverso wordnet.

8.2.2 Vantaggi

- **Dipendente dalla frase**: grazie alle relazioni tra le parole viene realmente tenuto conto del senso della frase, migliorando quindi l'affidabilità del sistema.

8.2.3 Svantaggi

- **Eccessivamente dipendente dal POS**: l'analisi si basa completamente sui risultati del POS, sia per le relazioni che per i TAG, nel caso di errori di questo componente (il cui risultati non è sempre affidabile) l'intera analisi è corrotta.

I vari componenti POS (spaCy, ClearNLP, CoreNLP, MATE, Turbo, SyntaxNet) hanno un affidabilità intorno al 93%[11], questo tipo di analisi è estremamente difficoltosa nel caso di frasi la cui sintassi non è perfettamente corretta o eccessivamente gergale.

Lo scopo del progetto è di rendere il più possibile flessibile il sistema, questo approccio avrebbe aumentato l'affidabilità, al prezzo di rendere il sistema funzionante solo per frasi con una sintassi perfetta. Abbiamo quindi scelto di togliere questo componente dal sistema.

8.3 doc2vec

8.4 word2vec

Capitolo 9

Modelli word2vec

Capitolo 10

Ricerca dei parametri

10.0.1 openie

Capitolo 11

Lattex Tutorial

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

- Elemento A
- Elemento B
- Elemento C

- Elemento A
- Elemento B
- Elemento C

1. Alpha
2. Beta
3. Gamma

[1]

`Questo testo ha una spaziatura fissa`
Questo testo è in italico
Questo testo è in grassetto
QUESTO TESTO È IN MAIUSCOLETTO

[1] Questa è una nota a pié di pagina.

Questo testo è sottolineato

Citazione:

> Donec et nisl id sapien blandit mattis. Aenean dictum odio sit amet risus. Morbi purus. Nulla a est sit amet purus venenatis iaculis. Vivamus viverra purus vel magna. Donec in justo sed odio malesuada dapibus. Nunc ultrices aliquam nunc. Vivamus facilisis pellentesque velit. Nulla nunc velit, vulputate dapibus, vulputate id, mattis ac, justo. Nam mattis elit dapibus purus. Quisque enim risus, congue non, elementum ut, mattis quis, sem. Quisque elit.

11.1 Sezione

Donec et nisl id sapien blandit mattis. Aenean dictum odio sit amet risus. Morbi purus. Nulla a est sit amet purus venenatis iaculis. Vivamus viverra purus vel magna. Donec in justo sed odio malesuada dapibus. Nunc ultrices aliquam nunc. Vivamus facilisis pellentesque velit. Nulla nunc velit, vulputate dapibus, vulputate id, mattis ac, justo. Nam mattis elit dapibus purus. Quisque enim risus, congue non, elementum ut, mattis quis, sem. Quisque elit.

11.1.1 Sotto sezione

Un po' di matematica:

$\frac{n!}{k!(n-k)!} = \binom{n}{k}$

Un po' di matematica centrata:

$$\frac{n!}{k!(n-k)!} = \binom{n}{k}$$

Oppure con $$

$$\frac{n!}{k!(n-k)!} = \binom{n}{k}$$

Oppure anche direttamente nel testo $\frac{1}{n}$

Donec et nisl id sapien blandit mattis. Aenean dictum odio sit amet risus. Morbi purus. Nulla a est sit amet purus venenatis iaculis. Vivamus viverra purus vel magna. Donec in justo sed odio malesuada dapibus. Nunc ultrices aliquam nunc. Vivamus facilisis pellentesque velit. Nulla nunc velit, vulputate dapibus,

vulputate id, mattis ac, justo. Nam mattis elit dapibus purus. Quisque enim risus, congue non, elementum ut, mattis quis, sem. Quisque elit.

Bibliografia

[1] Deep Learning. https://en.wikipedia.org/wiki/Deep_learning.

[2] Deng Li and Li Xiao. Machine Learning Paradigms for Speech Recognition:An Overview. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tasl-deng-2244083-x_2.pdf.

[3] Tomas Mikolov. Distributed Representations of Words and Phrases and their Compositionality. https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

[4] Agente Inteligente. https://it.wikipedia.org/wiki/Agente_intelligente.

[5] REpresentational State Transfer. https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm.

[6] Princeton University "About WordNet." WordNet. Princeton University. 2010. http://wordnet.princeton.edu.

[7] WordNet word similarity methods. http://search.cpan.org/~tpederse/WordNet-Similarity/.

[8] WordNet word path similarity. http://search.cpan.org/~tpederse/WordNet-Similarity/lib/WordNet/Similarity/path.pm.

[9] Part-of-speech tagging. https://en.wikipedia.org/wiki/Part-of-speech_tagging.

[10] Part-of-speech categories Treebank. https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html.

[11] Part-of-speech benchmarks. https://spacy.io.